

systemd Quick Reference

Service management, units, timers, and journalctl

Service Management

Basic Service Commands

```
systemctl start nginx
systemctl stop nginx
systemctl restart nginx
systemctl reload nginx # reload config
systemctl status nginx
```

Enable / Disable

```
systemctl enable nginx # start at boot
systemctl disable nginx # remove from boot
systemctl enable --now nginx # enable + start
systemctl is-enabled nginx
```

Service States

active (running)	Service is running normally
active (exited)	Ran once and exited successfully
inactive (dead)	Service is stopped
failed	Service crashed or exited with error
activating	Service is starting up

Unit Files

Unit File Location

/etc/systemd/system/	Admin-created units (highest priority)
/run/systemd/system/	Runtime-generated units
/usr/lib/systemd/system/	Package-installed units
~/.config/systemd/user/	User-level units

Basic Service Unit

```
[Unit]
Description=My Application
After=network.target
[Service]
ExecStart=/usr/bin/myapp --config /etc/myapp.conf
Restart=on-failure
User=appuser
[Install]
WantedBy=multi-user.target
```

Apply Changes

```
systemctl daemon-reload # reload unit files
systemctl restart myapp # apply changes
```

Timers

Timer Unit

```
[Unit]
Description=Run backup daily
[Timer]
OnCalendar=*-*-* 02:00:00
Persistent=true
[Install]
WantedBy=timers.target
```

OnCalendar Syntax

--* 02:00:00	Daily at 2:00 AM
Mon *-*-* 09:00:00	Every Monday at 9 AM
- 01 00:00:00	First day of every month
hourly / daily / weekly	Shorthand schedules

Timer Management

```
systemctl list-timers --all
systemctl start backup.timer
systemctl enable backup.timer
systemd-analyze calendar "daily"
```

Targets

Common Targets

multi-user.target	Normal boot, multi-user, no GUI
graphical.target	Full GUI desktop
rescue.target	Single-user rescue mode
emergency.target	Minimal shell, root only
network-online.target	Network is fully configured
timers.target	All timer units ready

Target Commands

```
systemctl get-default
systemctl set-default multi-user.target
systemctl isolate rescue.target
systemctl list-dependencies graphical.target
```

Journalctl

Viewing Logs

```
journalctl -u nginx # logs for unit
journalctl -u nginx -f # follow (tail)
journalctl -u nginx --no-pager
journalctl -b # current boot only
```

Filtering Logs

```
journalctl --since "2026-03-01"
journalctl --since "1 hour ago"
journalctl -p err # errors and above
journalctl _PID=1234
```

Priority Levels

emerg (0)	System is unusable
alert (1)	Immediate action needed
crit (2)	Critical condition
err (3)	Error condition
warning (4)	Warning condition
info (6)	Informational
debug (7)	Debug-level messages

Log Maintenance

```
journalctl --disk-usage
journalctl --vacuum-size=500M
journalctl --vacuum-time=30d
```

Networking

networkctl

```
networkctl list
networkctl status eth0
networkctl up eth0
networkctl down eth0
```

systemd-resolve

```
resolvectl status
resolvectl query example.com
resolvectl flush-caches
resolvectl statistics
```

Network Wait

```
# In unit file [Unit] section:
After=network-online.target
Wants=network-online.target
```

Mounts

Mount Unit

```
[Unit]
Description=Mount data volume
[Mount]
What=/dev/sdb1
Where=/mnt/data
Type=ext4
Options=defaults,noatime
[Install]
WantedBy=multi-user.target
```

Automount Unit

```
[Unit]
Description=Automount data on access
[Automount]
Where=/mnt/data
TimeoutIdleSec=300
[Install]
WantedBy=multi-user.target
```

Naming Convention

/mnt/data	Unit file: mnt-data.mount
/var/lib/app	Unit file: var-lib-app.mount

Mount path with ` ` replaced by `.` , leading dash removed

Environment

Setting Environment Variables

```
[Service]
Environment=APP_ENV=production
Environment=PORT=8080
EnvironmentFile=/etc/myapp/env
```

Environment File Format

```
# /etc/myapp/env
APP_ENV=production
DATABASE_URL=postgres://localhost/db
SECRET_KEY=changeme
```

Service Hardening

ProtectSystem=strict	Read-only filesystem except allowed paths
ProtectHome=true	Hide /home, /root, /run/user
NoNewPrivileges=true	Prevent privilege escalation
PrivateTmp=true	Isolated /tmp for the service
ReadWritePaths=/var/lib/myapp	Allow writes to specific paths

Dependencies

Ordering and Requirement Directives

After=b.service	Start after b (ordering only)
Before=b.service	Start before b (ordering only)
Requires=b.service	Hard dependency; fail if b fails
Wants=b.service	Soft dependency; don't fail if b fails
BindsTo=b.service	Stop when b stops
Conflicts=b.service	Cannot run at the same time as b

Inspecting Dependencies

```
systemctl list-dependencies nginx
systemctl list-dependencies --reverse nginx
systemd-analyze dot nginx.service | dot -Tsvg > deps.svg
```

systemd Quick Reference

Common Patterns

Restart Policies

Restart=no	Never restart (default)
Restart=on-failure	Restart on non-zero exit
Restart=always	Always restart (for daemons)
RestartSec=5	Wait 5 seconds before restarting
StartLimitBurst=3	Max restarts in interval
StartLimitIntervalSec=60	Interval for burst counting

Override Without Editing

```
systemctl edit nginx # creates drop-in
# /etc/systemd/system/nginx.service.d/override.conf
systemctl cat nginx # show effective config
systemctl revert nginx # remove overrides
```

System Analysis

```
systemd-analyze # boot time
systemd-analyze blame # per-unit time
systemd-analyze critical-chain
systemctl list-units --failed
```